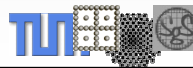


Principles of Mathematical Modelling

- describe a given problem with some **mathematical formalism** in order to
 - get a formal and precise description
 - see fundamental properties due to the abstraction
 - allow a systematic treatment and, thus, solution
- (mathematical) **model**: formal description (and usually simplification) of (some) reality
- bigger or smaller evidence:
 - exact natural science and engineering: long tradition (basic conservation laws of continuum mechanics, e.g.)
 - economics, game theory, climate modelling: many open questions (Keynes or not, MinMax or not, chaos or not?)
- to do: both *derivation* and *analysis* of models



Introduction to Scientific Computing
Lesson 3: Mathematical Modelling



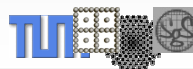
Slide 1

Derivation of a Model 1

- What do you want to model?
 - a catalyst's function or the detailed reactions in it?
- Which are the important quantities for that?
 - Example 1: Optimum trajectory of the Space Shuttle – gravitation of Pluto, gravitation of the Earth?
 - Example 2: Prediction of Dow Jones Index tomorrow – statements of Mr. Greenspan, statements of myself?
- How important are they?
 - Think of consequences of a neglect!
- What are their relations and interactions?
 - *qualitative* and *quantitative* aspects
- How can these be (mathematically) described?
 - algebraic or differential equations, graphs, automata, ...



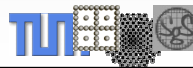
Introduction to Scientific Computing
Lesson 3: Mathematical Modelling



Slide 2

Derivation of a Model 2

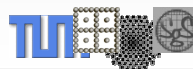
- description of relations and interactions:
 - position, speed, and acceleration of an oscillating pendulum?
ordinary differential equation!
 - deformation of a membrane under some load?
partial differential equation!
 - initial or boundary conditions of some growth process?
algebraic equations!
 - non-negativity of some quantity?
algebraic inequality!
 - order of several steps?
graphs!
 - state transitions?
automata!



Derivation and Analysis of Models

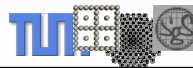
- Of which type is the resulting task?
 - Is there a solution (Hamiltonian way in a graph)?
 - Find a/the solution (flow field around an aircraft)!
 - Find a/the best solution (shape optimization)!
- What can be said about solution(s) concerning
 - their existence?
 - their uniqueness?
 - their dependency on the input data?

(*well-posed problems*: Hadamard 1923; Tikhonov, John; cf. *inverse problems*)
- Is the model well-suited for a numerical treatment?
- Is the model derived so far correct?
 - validation (experiments)!



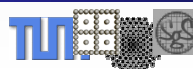
What to do with models?

- the **analytical** approach:
 - prove existence and uniqueness formally
 - construct or find solution(s) formally/directly/analytically
 - desirable, but almost never possible
- the **heuristic** approach:
 - trial and error, following some (hopefully smart) strategy
 - useful in discrete problems (travelling salesman etc.)
- the **direct numerical** approach:
 - follow some numerical algorithm and end up with the exact solution (Simplex algorithm for linear programming)
- the **approximative numerical** approach:
 - approximate/discretize the model equations and end up with some approximate solution



Classes of Mathematical Models

- Models can be **discrete** or **continuous**:
 - discrete models use a discrete/combinatoric description (integer numbers, graphs, ...)
 - continuous models use real quantities (real numbers, physical quantities, differential equations, ...)
 - primarily, but not necessarily: discrete models for discrete phenomena, continuous models for continuous phenomena
 - examples: lattice-gas-automata for fluid flow, continuum mechanics for traffic flow
- Models can be **deterministic** or **stochastic**:
 - again no general relation between phenomena and models
 - roll the die: random phenomenon, stochastic model
 - crash test: deterministic phenomenon, deterministic model
 - but what about weather or data traffic through the internet?

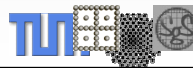


Discrete Models 1: Scheduling

- n jobs to be done on m machines working in parallel:
 - no job simultaneously on more than one machine
 - no machine working simultaneously on more than one job
- several model parameters:
 - characterizing the **jobs**: subjobs, processing time, earliest starting time, due time, weighting, cost of delay, ...
 - characterizing **machines** and **processing**: identical or different machines, order of subjobs important or not, precedence relations of jobs
 - criteria of **optimality**: time of completion, delay, idle times
- task: find the best schedule with respect to some objective function (minimizing overall time, e.g.)
- model: graph with disjunctive and conjunctive edges



Introduction to Scientific Computing
Lesson 3: Mathematical Modelling



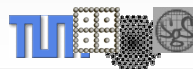
Slide 7

Discrete Models 2: Elections

- m candidates, n voters; model: theory of relations
- task: derive a reasonable *collective* preference from the different reasonable *individual* preferences
- possible choices:
 - external dictator: individual preferences do not matter ☹
 - internal dictator: one individual preference always wins ☹
 - majority wins: cycles ($a > b > c > a$) cannot be excluded ☹
 - something else democratic – but what is democratic?
 - all reasonable individual preferences are allowed
 - the collective preference must be reasonable
 - everything is possible with unanimity
 - no dictator, independence of irrelevant alternatives
- impossible if $m > 2$ and $n > 1$ (Arrows, 1951)!!
- drawback of democracy or drawback of our model??



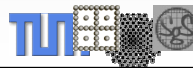
Introduction to Scientific Computing
Lesson 3: Mathematical Modelling



Slide 8

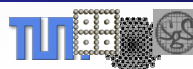
Discrete Models 3: Event Simulation

- task: model data or job traffic through computer system, optimize flow, find and avoid bottlenecks
- model: discrete event simulation, stochastic processes
 - elementary queueing system (service stations, waiting pool)
 - quantities: waiting time, service time, staying time; filling, capacity, throughput
 - arrival and completion of jobs: stochastic processes (deterministic D, Markovian M, or general G)
 - notation: M|M|1 (arrival and completion neg. exp., 1 station)
 - service strategies: FCFS, LCFS, random, round robin, ...
 - queueing nets: network of elementary queueing systems
- computer systems: discrete space of states, state transitions driven by stochastic processes, Markovian chains or systems, resp.



Hierarchy and Multiscale Property

- often to be chosen: scale/level of observation:
 - Which resolution is necessary (w.r.t. the model's accuracy)?
 - Which resolution can be tackled numerically?
- flow through a cylinder – how many dimensions?
 - 1D: neglect cross-section
 - 2D: exploit symmetries
 - 3D: full resolution
- electric circuit simulation – spatial resolution or not?
 - std. system simulators (SPICE, TITAN): only time, ODEs
 - parasitic cross effects: take space into account!
- turbulence – which vortices can be neglected?
 - significant transport of energy between different scales
 - direct simulation – Large Eddy Simulation – averaging models



Averaging and Homogenization

- often: coarse-grain phenomena are of interest, but fine-grain phenomena must not be neglected
- try to do some **averaging**:
 - in time: turbulence, molecular dynamics
 - in space: flow and transport through porous media (a catalyst or soil)
- formal concept: **homogenization**
 - representative elementary volume
 - scaled reproduction, translation, periodic continuation
 - limit process of scaling factor
 - new quantities (effective parameters: porosity, permeability)
 - new equations (porous media: instead of transport equations now Darcy-Forchheimer equation)

